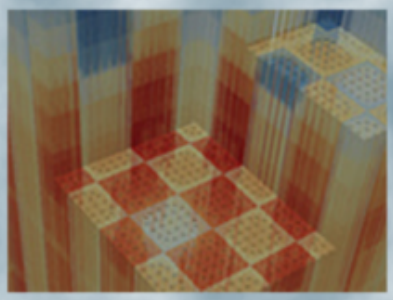
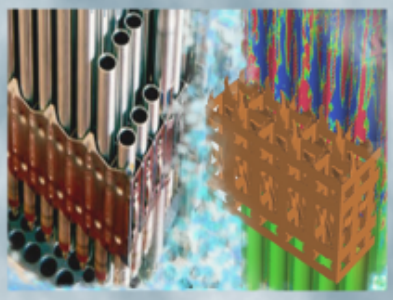
CASL-U-2016-1030-000





Development and Testing of CTF to Support Modeling of BWR Operating Conditions

Robert Salko, Oak Ridge National Laboratory
Aaron Wysocki, Oak Ridge National Laboratory
Benjamin Collins, Oak Ridge National Laboratory
Andrew Godfrey, Oak Ridge National Laboratory
Chris Gosdin, Pennsylvania State University
Maria Avramova, North Carolina State University

01/29/2016

REVISION LOG

Revision	Date	Affected Pages	Revision Description
0	05/15/15	All	Initial Release

Document pages that are:

Export Controlled _____

IP/Proprietary/NDA Controlled_____

Sensitive Controlled_____

Unlimited_____All_____

Requested Distribution:

To: CASL-SLT

Copy:

EXECUTIVE SUMMARY

This milestone supports developing and assessing COBRA-TF (CTF) for the modeling of boiling water reactors (BWRs). This is achieved in three stages. First, a new preprocessor utility that is capable of handling BWR-specific design elements (e.g., channel boxes and large water rods) is developed. A previous milestone (L3:PHI.CTF.P12.01) led to the development of this preprocessor capability for single assembly models. This current milestone expands this utility so that it is applicable to multi-assembly BWR models that can be modeled in either serial or parallel. The second stage involves making necessary modifications to CTF so that it can execute these new models. Specifically, this means implementing an outer-iteration loop, specific to BWR models, that equalizes the pressure loss over all assemblies in the core (which are not connected due to the channel boxes) by adjusting inlet mass flow rate. A third stage involves assessing the standard convergence metrics that are used by CTF to determine when a simulation is steady-state. The final stage has resulted in the implementation of new metrics in the code that give a better indication of how steady the solution is at convergence. This report summarizes these efforts and provides a demonstration of CTF's BWR-modeling capabilities.

(This page intentionally left blank)

CONTENTS

FIGURES

TABLES

ACRONYMS

3D three-dimensional

BFBT BWR Full-size Fine-mesh Bundle Tests

BWR boiling water reactor

CASL Consortium for Advanced Simulation of Light Water Reactors

CTF COBRA-TF

HDF5 Hierarchal Data Format 5

PETSc Portable Extensible Toolkit for Scientific Computations

PWR pressurized water reactor

QOI Quantity of Interst

T/H thermal-hydraulic

VERA Virtual Environment for Reactor Applications

VERA-CS Virtual Environment for Reactor Applications—Core Simulator

VTK Visualization ToolKit

XML Extensible Markup Language

1 INTRODUCTION

It is intended that Phase II of the Consortium for Advanced Simulation of Light Water Reactors (CASL) will add boiling water reactor (BWR) modeling capabilities to the Virtual Environment for Reactor Applications—Core Simulator (VERA-CS). The BWR design introduces several new and unique modeling and simulation challenges that will need to be addressed in the underlying code packages. The COBRA-TF (CTF) thermal-hydraulic (T/H) model is a two-phase, separated-flow model with constitutive models for the flow regimes that will be encountered in normal BWR analysis. Preparing CTF to be a BWR core simulator will involve developing a way to generate high-fidelity (pin-resolved) models for the code, testing and improving code performance running these models, and validating the code models that are significant to BWR operating conditions.

This document addresses the issues of creating the CTF models and performing some limited testing of the code's BWR-modeling capabilities. The validation work will be a more significant effort that is covered in future milestones. A new preprocessor utility is developed to create CTF native input decks from the Virtual Environment for Reactor Applications (VERA) common input (discussed in Section 2). This utility is used to create a mock BWR/4 model with nonuniform axial and radial power on the assembly level that is run in CTF (discussed in Section 3). Section 4 discusses a proposed set of new convergence criteria that should help to give a better sense of how steady the solution is when converged.

2 BWR PREPROCESSOR DEVELOPMENT

CTF models used in VERA-CS simulations are created at the pin-resolution level. As a result, the models often contain tens of thousands of channels and rods and thus, millions of mesh cells for core-scale models. All of these model entities must be explicitly defined in the native CTF input deck, which becomes very large as a result. Because manually creating such large models would be impractical, CASL has developed a separate preprocessor utility that generates native CTF input decks from the reduced, user-friendly input of the VERA common input file [1]. Originally, the preprocessor was developed for creating pressurized water reactor (PWR) models only. With CASL moving into BWR-modeling applications in Phase II, it is necessary to either extend the existing preprocessor to include BWR-specific designs or create a new preprocessor utility. A review of the PWR preprocessor in Milestone L3:PHI.CTF.P12.01 concluded that, due to the rigid design of the PWR preprocessor as well as several design shortcomings caused by information that was unavailable during its initial development, it is more feasible to develop a new utility for BWR applications.

2.1 Requirements

The requirements for the BWR preprocessor include the following:

1. Create a native CTF input deck from the VERAIn Extensible Markup Language (XML) file generated according to the VERAIn BWR input specification [2].
2. Allow for any number of different assembly types (including lattices of different sizes) in a model.
3. Allow for any number of stacked lattices in the assembly (the preprocessor should select the correct one to include in the CTF model).
4. Allow for any number of different pin types in a model.
5. Allow for rods with gadolinia (because CTF does not model this, the preprocessor should treat it by merging it into the pellet).
6. Allow for the presence of burnable absorber rods.

7. Consider the presence of large water rods (taking up four cells in the fuel lattice).
8. Create either serial or parallel versions of CTF input decks.
9. Consider the presence of partial-length rods.
10. Allow for the presence of guide tubes with varying axial dimensions.
11. Directly read the XML file as input (as opposed to the c++ front-end used by the PWR preprocessor).
12. Apply an assembly entrance loss to all BWR assemblies from an orifice loss coefficient map read from the VERAIn common input.

This set of requirements represents what the BWR preprocessor should be able to do when complete; however, the entire list has not yet been realized as part of this milestone. Part-length rods are currently not supported, and there is no treatment for guide tubes with varying dimensions.

The BWR preprocessor has been designed to model the active region of the core; the all axial lattices will be read from the XML file but only the lattice with fuel will be used to set up the CTF model. Any spacer grids outside the active region are ignored in the CTF model. This simplification will need to be revisited in the future, as the inlet and outlet flow losses may be different for different fuel assemblies, which will affect the assumption of a uniform outlet pressure boundary condition and the assumption that core pressure drop is the same over all assemblies.

2.2 Design

The preprocessor involves four main parts: (1) directly parse the VERAIn XML file using the ForTeuchos library, (2) build a raw "VERAIn" data structure that is very similar in structure to the XML file, (3) process the raw VERAIn data structure to create a "VERAInCTF" data structure that contains the model data relevant to a CTF model, and (4) produce the native CTF input deck. The dependency on the ForTeuchos library is an optional one, meaning the user will not get a BWR preprocessor executable after the build if the ForTeuchos library is not present. This is a drawback for users not wanting to satisfy these additional code dependencies; having a more lightweight XML parser that could be included in the CTF repository would be advantageous.

Assumptions about how to handle geometry issues such as the presence of burnable poisons and the location of the active region are made in type-bound procedures in the raw VERAIn data structure. This was done to make the preprocessor more flexible in handling currently unforeseen modeling issues like the ones that have been encountered with the PWR preprocessor.

The largest component of the BWR preprocessor source is the section that creates the VERAInCTF data structure. The data structure is designed with a structure similar to that of the VERAIn XML file itself. Figure 1 gives a general overview of the design of this data structure and its subcomponents.

The main data structure is the VERAInCTF class, which houses the CASEID object (not shown), the STATE objects (one for each STATE block in the VERAIn XML file), the EDITS object, and the COBRATF object, which are very similar in structure to the VERAIn file blocks with the same names. The Core class contains a list of pointers to unique Assem objects in the core. The SquareCore extension contains core maps (2D arrays) showing how assemblies are organized for that configuration. The Assem class is an abstract class containing a list of pointers to unique Pin objects in the assembly, a list of pointers to Channel objects, and GridCTF objects, which represent spacer grids. The Assem class is extended by a SquareLattice class, which contains data that are specific to square rod lattices (e.g., pin and channel maps). The SquareLattice class is extended by PWRAssem and BWRAssem classes. The BWRAssem class adds information about the channel box that surrounds the assembly.

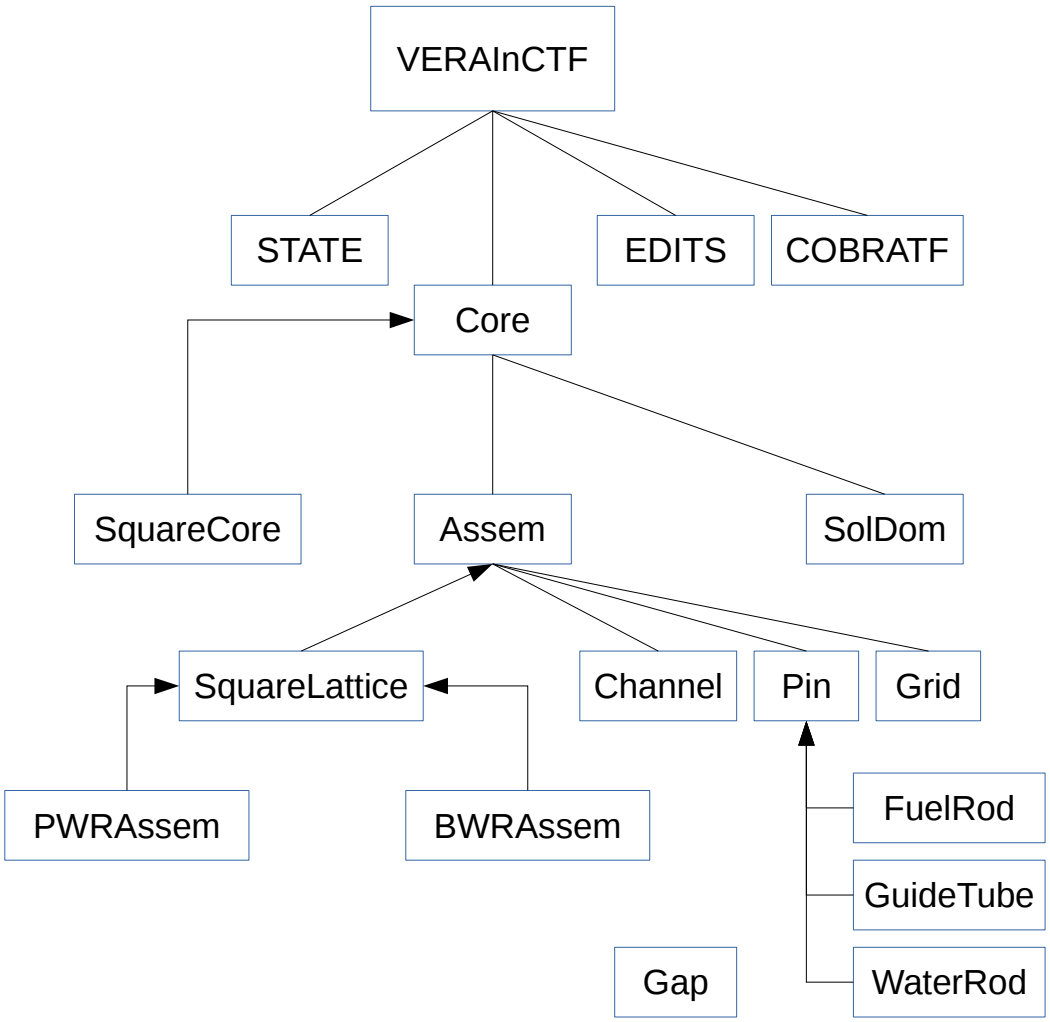


Figure 1. Design of VERAInCTF data structure

The Pin class has extensions for modeling fuel rods (FuelRod class), guide tubes (GuideTube class), and larger water rods found in BWRs and some PWR designs (WaterRod class). A gap, which is a connection between two adjacent channels in the model, also has a class. The class contains two pointers that point to the two channel objects linked by the gap.

All actual Pin, Channel, Assem, and Gap objects are stored in lists not owned by any class. It is not intended that any of these objects should be directly accessed from these lists, which act as a sort of repository for the data. Rather, pointers to these objects are stored by classes that own them. This prevents the duplication of unique objects in memory and allows the code to easily pass pointers around that can be used for accessing data from the objects. For example, an Assem object may own a few hundred pins; however, there may only be a few unique pin types in the assembly. In this case, there are a few Pin object extensions in the pin repository list, and the Assem object owns the several hundred pointers to those actual objects in the repository list. When the deck writer procedure needs information about those pins, it receives the pointers to the objects, which are then used to access the pin data for writing it to the CTF input deck.

The SolDom class is also owned by the Core class. A SolDom is a collection of entities (pins, channels, and gaps) found in one solution domain of the model. A serial case will have only one SolDom object while a parallel one will have multiple solution domains. The SolDom objects are built after the model (pin and channel layout and how everything connects) is set up. The SolDom object will collect the Pin, Channel, and Gap pointers from their owner classes in the Core object as well as add additional information for the entities related to parallel runs (e.g., whether the entity is a ghost and what other entities ghost it if not).

When it comes time to write the CTF input deck, all entities are written based on the solution domain, meaning that the deck writer procedure is not directly talking to the objects in the Core class, but to the objects in the SolDom class. This design keeps the physical model separate from the domain decomposition algorithm used for parallelizing the code and should allow for relatively easy modifications to the algorithm in the future should there be an interest in reducing the size of solution domains or allowing for more flexible processor configurations. Currently, the domain-decomposition is tied to the physical number of assemblies in the model as is the case with the PWR preprocessor.

This preprocessor was designed in a general, flexible way so that it is possible to extend its capabilities to modeling PWR core designs. Because several shortcomings of the current PWR preprocessor are addressed by this new utility, this may be a desirable endeavor. In addition to expanding the capabilities of the PWR preprocessor, this would be better for the user (not requiring two different utilities for making a CTF input deck) as well as the developer (not requiring maintenance and testing of two different utilities).

The PWRAssem class data and procedures are present but not completely functional. While the class will properly construct all pins in the assembly and channels within the pin lattice, there is currently no means of creating the inter-assembly channels that link all assemblies together.

2.3 Testing

The preprocessor has been tested by designing a small collection of BWR single- and multiple-assembly models using the VERAIn common input, running the BWR preprocessor to produce the CTF input deck(s), and then hand-verifying that the models are correct. These models were then used to set up regression tests for the BWR preprocessor, using the verified CTF input deck as the gold file. The tests require zero textual differences for the tests to pass. All BWR tests live in the CTF Github repository; this is a benefit over the PWR preprocessor, which has its tests split between the CTF repository and the VERAIn repository due to the need for the c++ front-end XML processor. BWR preprocessor regression tests are summarized in Table 1.

Table 1. BWR regression tests

Number	Ctest name	Description
1	COBRA_TF_bwrpreproc_bwr-peach-6	Single-assembly BWR 8×8 assembly with two single-cell guide tubes
2	COBRA_TF_bwrpreproc_cell_large	Single-assembly BWR 8×8 assembly with one large water rod in center
3	COBRA_TF_bwrpreproc_bwr_4x4_assem	Multiassembly (4 8×8 BWR assemblies) with two unique fuel assembly types
4	COBRA_TF_bwrpreproc_par_2x2_assem_par	Multiassembly model (same as 3) built to run in parallel

As an added feature, the BWR preprocessor will write a post-script file that contains a top-view picture of the core. These images are shown for Test number 1, 2, and 3 of Table 1 in Figures 2, 3, and 4, respectively. The post-script feature does not render the BWR channel box corners correctly. The corners show up as square instead of round in the picture; however, the flow area and wetted perimeter are correct in the CTF model.

Axial information is the same between the four tests. All tests use the same mesh as the Problem 6 test problem [3], which places one level for each spacer grid (equal to the height of the spacer grid), with other axial levels between the grids being about 7.6 cm (3 in.) in height. There are 49 axial levels in the active region of the model. There are three lattices that comprise the assembly in the VERAIn file: an inactive gap between lower plenum and the active region of the fuel, an active fuel region, and an upper plenum section with no active fuel. The multiple axial lattices are placed in the file to ensure that the preprocessor picks the correct region to model in CTF. These tests contain only one lattice with fuel; models containing multiple fuel lattices will require additional logic in the preprocessor for picking the correct one (or handling merging them). Additionally, eight spacer grids were placed in the models with one grid being above the active length, which tests that the preprocessor correctly ignores the upper grid.

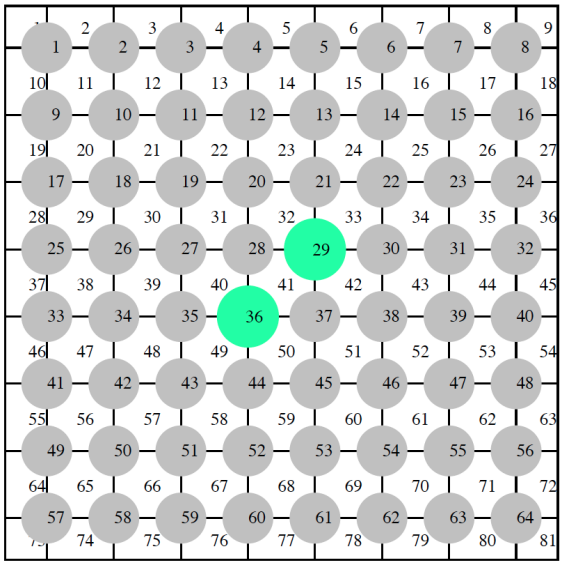


Figure 2. Diagram of BWR preprocessor regression test, COBRA_TF_bwrpreproc_bwr-peach-6

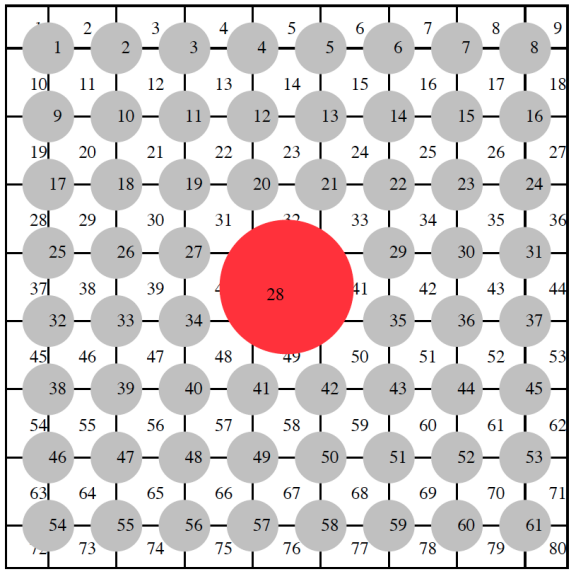


Figure 3. Diagram of BWR preprocessor regression test, COBRA_TF_bwrpreproc_cell_large

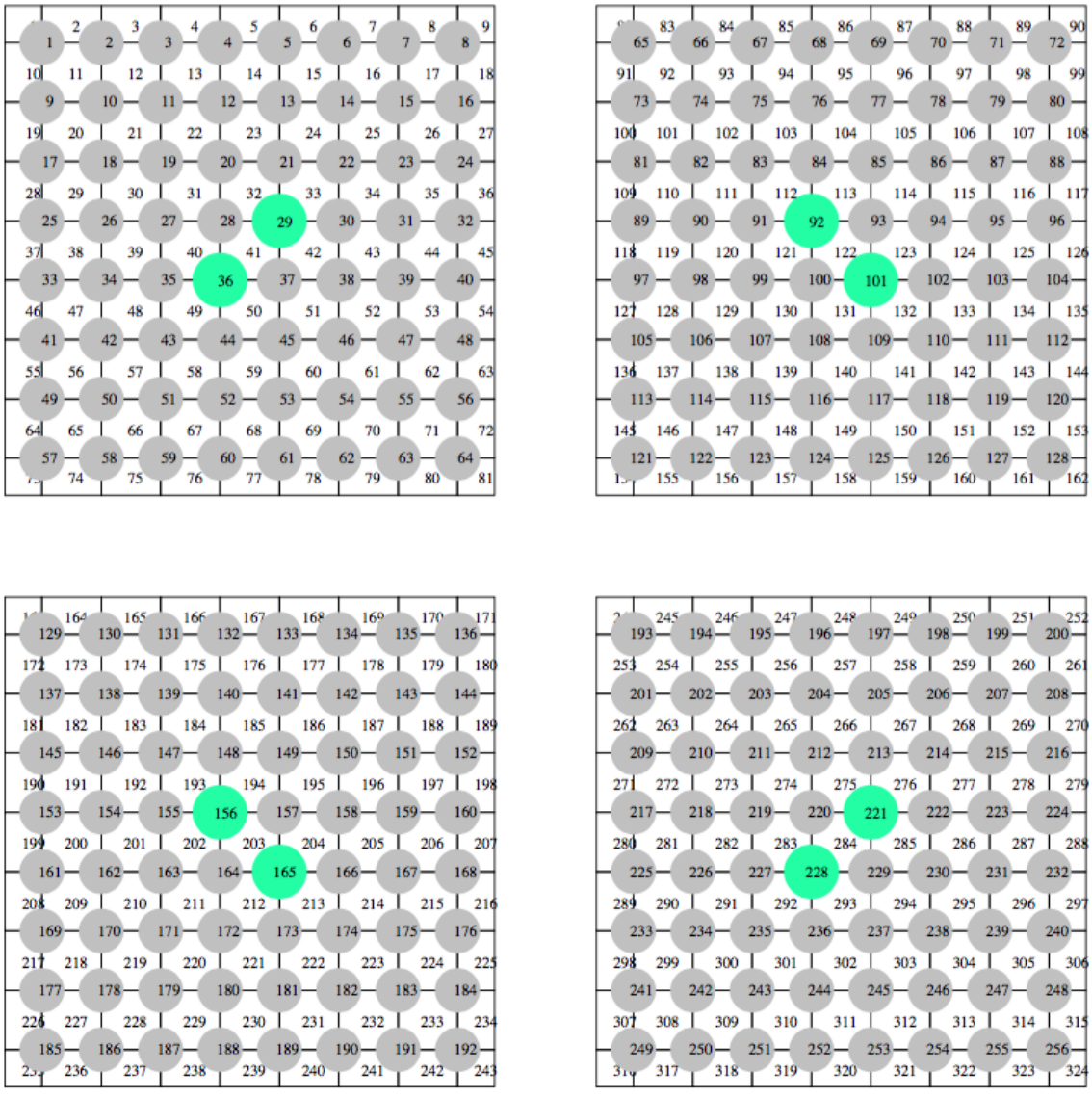


Figure 4. Diagram of BWR preprocessor regression test, COBRA_TF_bwrpreproc_bwr_4x4_assem

3 CTF OUTER ITERATION LOOP

3.1 Algorithm Description

The assemblies in the multiassembly models produced in Table 1 are completely separate from one another due to the presence of the channel boxes. The models do not connect at the inlet or outlet as there is no lower or upper plenum included in the CTF model. The CTF design actually prevents the user from creating a lower plenum for one of those models due to a hard-coded limit on how many channels may connect between section boundaries (a separate axial section would be required to model the lower plenum, and the lower plenum section channels would need to be connected to the core section channels). While it is possible to remove this hard-coded restraint, it is unclear how difficult it will be for the code to converge the pressure matrix when there is one large channel at the bottom of the model that connects all channels in the core (a pressure equation with tens of thousands of terms). Furthermore, the CTF parallelization approach makes the assumption that the model only contains one section and so modification would be required.

A more straightforward solution, which requires no change to the parallel solution algorithm and has a higher certainty of success, is to implement an outer iteration loop in the code. The method works by checking if the model is for a BWR during the steady-state convergence check that is done at the end of every CTF time step. If the solution is considered to be steady-state and the model is a BWR, the code checks if the global core pressure drop is converged. If the pressure drops in all assemblies do not match to within a specified tolerance, the code calculates new inlet mass flow rates and sets them as the new inlet boundary conditions in an attempt to match the pressure drops in all assemblies. Another CTF steady-state solution will be performed and the global pressure drop will again be checked. If all assembly pressure drops are within 0.7 kPa (0.1 psi) of one another, the outer pressure iteration loop is considered converged and the solution exits. This process is summarized in the simple flow chart in Figure 5.

The step that calculates the new flow distribution works by generating a simple linear relationship between inlet mass flow rate and bundle pressure drop using the results of two successive outer iterations. The first outer iteration of the simulation makes a small perturbation to the bundle inlet mass flow rate to generate the data for the linear relationship. This currently works by checking if the bundle pressure drop is higher or lower than the core average pressure drop. If higher, the inlet mass flow rate is reduced by 5 % or, if lower, the inlet mass flow rate is increased by 5 %. After the second CTF solve, the linear relationship is built from the two sets of inlet mass flow rate and bundle pressure drop. Successive outer iterations continue to update the relationship with new data. The equation is of the form shown below:

$$\Delta P_b = C_{0,b} + C_{1,b}\dot{m}_b \tag{1}$$

Where C_0 and C_1 are the coefficients of the correlation. The b subscript means "bundle," indicating that one correlation is built for each fuel assembly in the model. ΔP_b is the pressure drop over the bundle and $\dot{m}$ is the inlet mass flow rate of the bundle. This can be rearranged so it gives bundle mass flow rate for a given pressure drop:

$$\dot{m}_b = \frac{\Delta P_b - C_{0,b}}{C_{1,b}} \tag{2}$$

The sum of the individual assembly inlet mass flow rates should sum to the total specified core inlet mass flow rate. Additionally, the pressure drop over the core should be equal in all assemblies. These two facts allow the sum of the individual bundle equations to be used to predict the final core pressure drop.

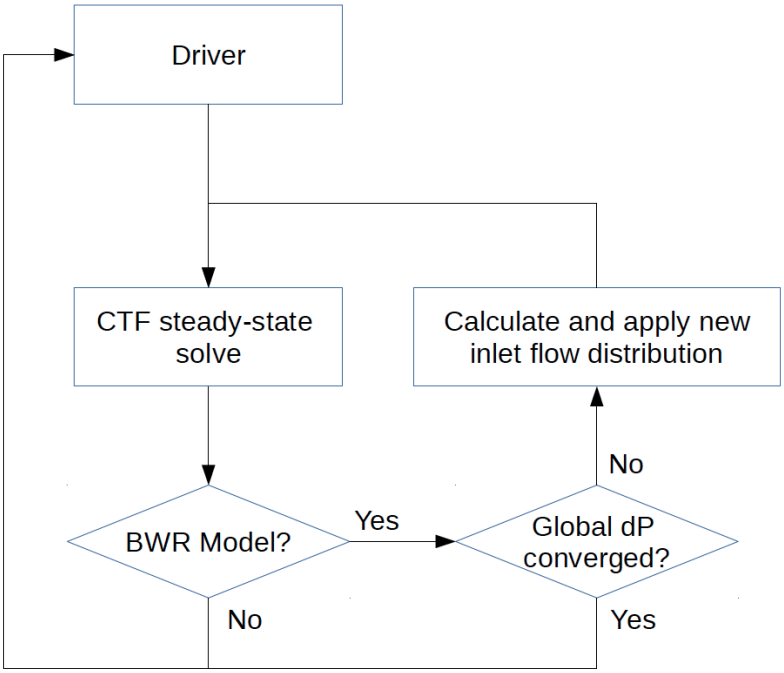


Figure 5. Flowchart of the outer iteration loop that has been implemented in CTF for BWR models

$$\Sigma_{b=1}^{B}\dot{m}_b = \dot{m}_{\text{core}} = \Sigma_{b=1}^{B}\frac{\Delta P_{\text{core}} - C_{0,b}}{C_{1,b}} \tag{3}$$

Where B is the total number of assemblies in the core, $\dot{m}_{\text{core}}$ is the total active flow into the model, and ΔP_{core} is the predicted final pressure drop over the core. Solving for the core pressure drop yields the following:

$$\Delta P_{\text{core}} = \frac{\dot{m}_{\text{core}} + \Sigma_{b=1}^{B}\frac{C_{0,b}}{C_{1,b}}}{\Sigma_{b=1}^{B}\frac{1}{C_{1,b}}} \tag{4}$$

After the final total core pressure drop is calculated, this term is simply substituted back into Eq. 2 to get the predicted required bundle inlet mass flow rate to produce this pressure drop.

3.2 Additional Modifications

A few minor additional changes of CTF were required for BWR modeling. CTF needs to know it is modeling a BWR to perform the global pressure outer iteration loop. A new flag was added to Card 1 of the native CTF input deck called "BWRMODEL." When set to "1," CTF will assume it is modeling a BWR and will do a few things differently (discussed below). The CTF User Manual has been updated with this change [4].

First, CTF will implement the algorithm shown in Figure 5 when modeling a BWR. Second, the code will default to using the serial biconjugate gradient stabilized solver that is included with CTF (Sparskit2 library) for doing the pressure matrix solve. The Portable Extensible Toolkit for Scientific Computations (PETSc) solvers that are included in CTF for parallel runs are not used for BWR models because they have been implemented in a way that builds one giant global pressure matrix for the entire system, which is necessary for a PWR model, where everything is connected. However, in a BWR model, there are no connections between different assemblies, and so using the PETSc solvers in their current form would only lead to the overhead of building one giant matrix and iteratively solving it. Modifying the PETSc solvers to build individual pressure matrices for each assembly should be considered in the future.

Third, CTF will default to using the new convergence criteria that are discussed in Section 4. It has been found during testing that it is important to sufficiently converge each CTF steady-state solve prior to adjusting the mass flow rates to generate good data for the $\dot{m}/\Delta P$ model. Suitable values for judging convergence have been found for the test case using the new convergence criteria. However, these defaults may need to change in the future (based on problem conditions).

3.3 Testing

The global pressure outer iteration loop feature has been tested by building BWR core models of varying sizes using the BWR preprocessor. Initially, small cores on the order of 4–24 8×8 assemblies were created to allow for testing on the CASL Fissile machines. After the algorithm was shown to work satisfactorily, the problem size was scaled up to something more representative of a BWR/4 design.

Figure 6 shows how bundle pressure drop responds to inlet mass flux for one of the small cores of 8×8 bundles with nonuniform radial power distribution solved using the algorithm. The figure shows pressure drop and mass flux behavior for three selected assemblies of different power levels. This problem converges in four outer iterations. The flow is initially uniform at the inlet; during the first iteration, the three assembly mass flow rates lie on a vertical line. The pressure drops are not the same because of the difference in the assembly powers. The flow is adjusted by a small amount (5 %) after the first iteration, which leads to a small change in pressure drop over the second iteration. At the completion of the second iteration, the linear relationship is formed, and the mass flux is changed by a much larger amount, leading to a pressure drop that is much closer to a single core average at the end of the third iteration. The final iteration is used to push the pressure drops to be within the set tolerance. The final state of the core involves the inlet mass flux of the three assemblies being very different, but the pressure drops all being very similar.

An important takeaway from Figure 6 is the relationship between the mass flux and bundle pressure drop, which is quite linear. While the frictional pressure drop will decrease quadratically with velocity, there is a void feedback effect that prevents the two-phase pressure drop from exhibiting this same behavior. As inlet mass flow rate is reduced, overall void content increases due to the mixture enthalpy increasing and possibly also due to the slip ratio reducing. The increase in void increases the two-phase multiplier on the single-phase pressure drop, leading to the linear relationship observed in the figure.

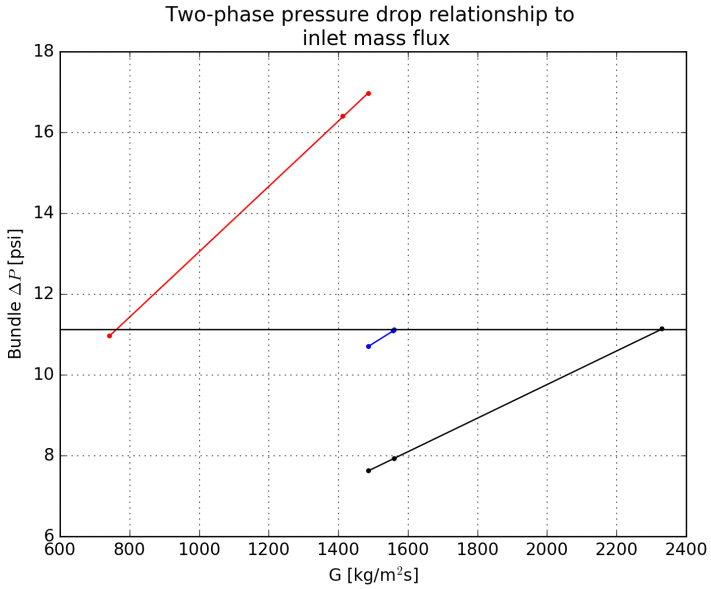


Figure 6. Change in mass flux predicted by outer iteration loop for select assemblies in BWR mini-core problem and bundle pressure drop response

A full core model with 764 assemblies was created to further test the algorithm. The model contains roughly 61,000 subchannels, 3 million fluid mesh cells, and 48 million solid mesh cells. The radial power distribution was made nonuniform on the assembly level; the radial power distribution within an assembly was uniform (with the exception of the guide tubes, which had zero power). Figure 7 shows the assembly peaking factors for the model. The figure only shows the peaking factors for a quarter of the model, but the full core was actually modeled. The axial power distribution was a modified cosine shape that was adjusted to be more bottom peaked to represent the effect of large amounts of void in the upper portion of the core. Figure 8 shows the axial power shape that was applied to all rods in the model. Information that was used in constructing the model is shown in Table 2. The assembly that was shown in Figure 2 was used for all assemblies in the mock BWR/4 model.

The model took five iterations of the pressure loop to converge on a single core pressure drop. Table 3 shows a summary of the convergence behavior for the simulation. The first perturbation of the inlet mass flow rate was 5 %, which was apparently much too small of a change to have a significant effect on the bundle pressure drop. After this small perturbation generated data for the linear curve fit, the third iteration led to a much bigger change in inlet mass flow rate and bundle pressure drop, requiring a few hundred extra iterations to converge the solution. The requirement for convergence of each outer iteration loop was for the void, pressure, fluid temperature, solid temperature, and liquid velocity l_2-norms to be less than $1.0{\cdot}10^{-4}$ and the vapor and droplet velocities to be less than $1.0{\cdot}10^{-3}$ (see Section 4 for details about the new CTF convergence metrics).

The Visualization ToolKit (VTK) file was used to generate some three-dimensional (3D) visualizations of the fluid data. Figure 9 shows the inlet mass flux distribution from the converged case. Despite starting the simulation with a uniform mass flux distribution, the flow became very non uniform by the end of the simulation. Flow migrated to the core periphery due to the higher void and flow resistance in the center. In a real BWR, assembly entrance loss coefficients would be adjusted to prevent too much flow from leaving the core center.

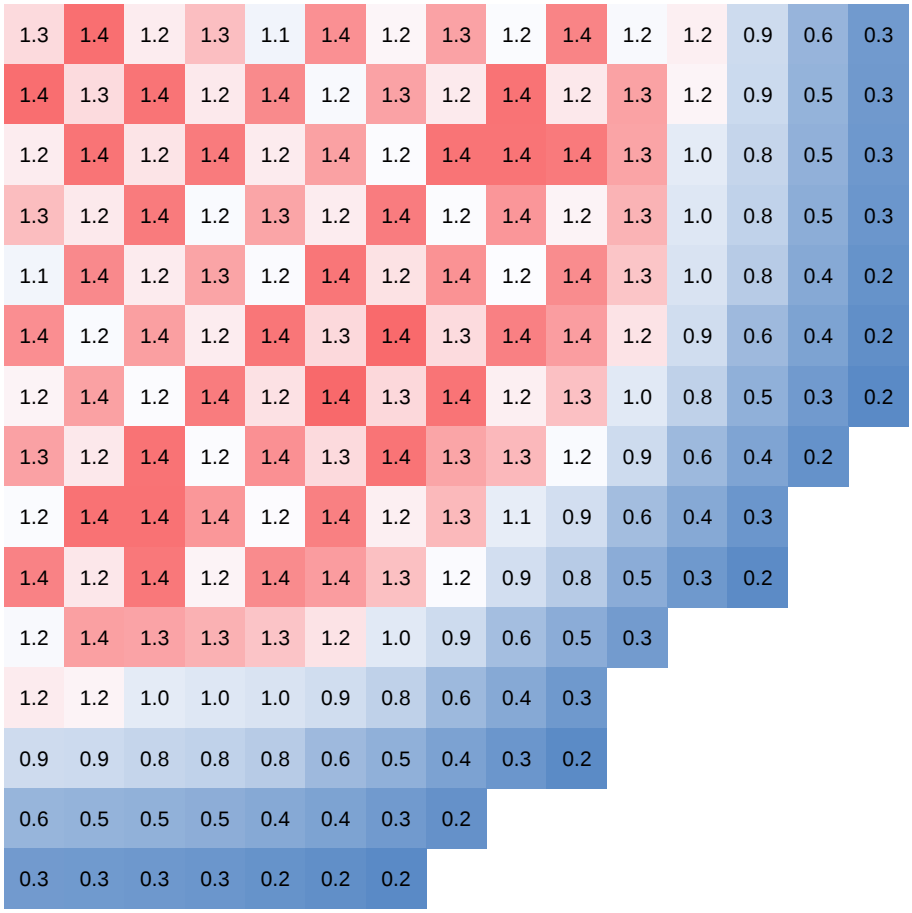


Figure 7. Assembly peaking factors for mock BWR/4 model (only quarter of model shown)

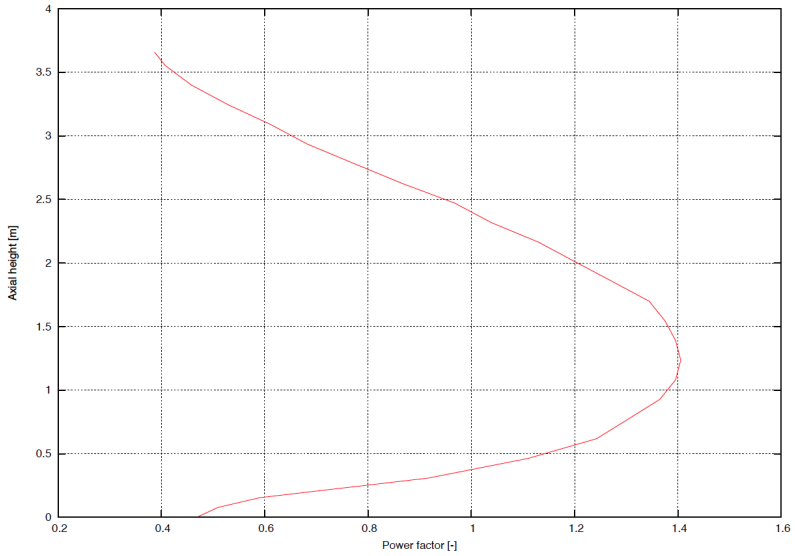


Figure 8. Axial peaking factor shape for all rods in mock BWR/4 model

Table 2. Mock BWR/4 model data

Parameter	Value
Inlet active flow	12 200 kg s^{-1} (96.82 Mlbm h^{-1})
Rated power	3514 MWt
Gamma heating	2 %
Outlet pressure	71.70 bar (1040 psia)
Inlet temperature	276.9 °C
Number of assemblies	764
Assembly pitch	15.24 cm
Core active length	365.76 cm
Assembly size	8×8
Pin pitch	1.6256 cm
Channel box radius	0.254 cm
Number of guide tubes per assembly	2
Fuel rod radius	0.613 41 cm
Guide tube radius	0.750 57 cm
Number of spacer grids	7
Grid loss coefficient	0.9070
Number of axial levels	49
Number of subchannels	61,884
Fluid mesh cells	3,032,316
Solid mesh cells	48,938,784

Table 3. Outer iteration convergence behavior for the mock BWR/4 simulation

Outer iteration	Number of inner iterations	Max ΔP from core average (psi)
1	2097	5.69
2	1720	5.29
3	2063	1.80
4	1541	0.33
5	1312	0.02

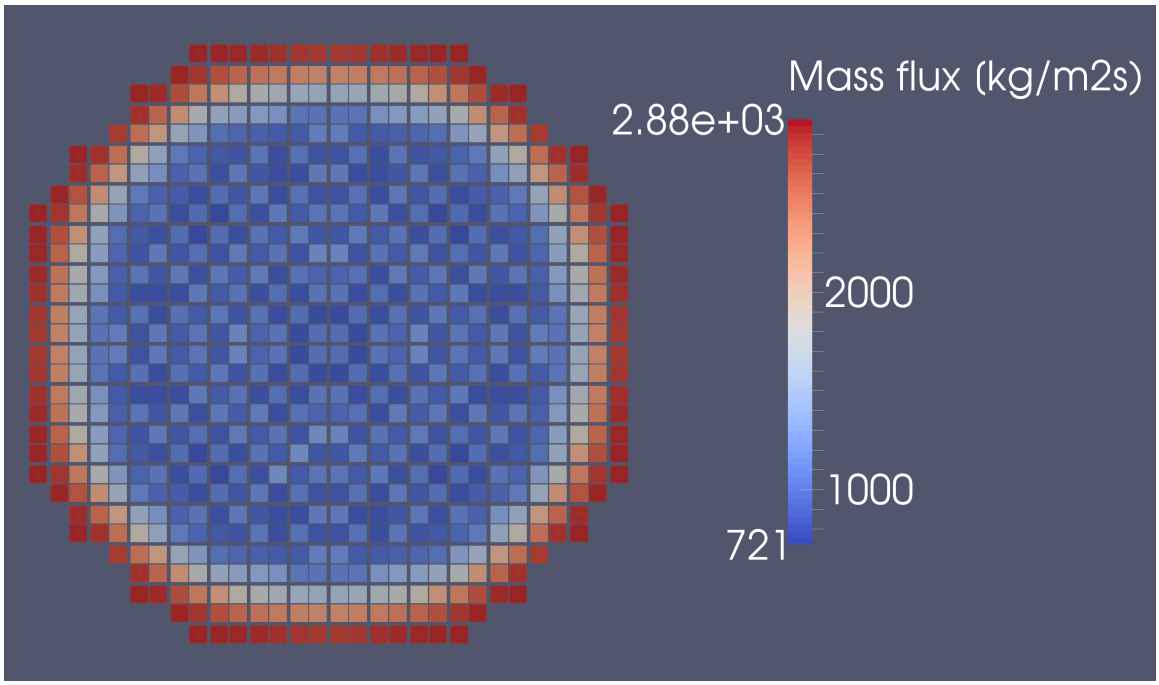


Figure 9. Core inlet mass flux distribution

The outlet void fraction is shown in Figure 10. It is clear from the figure that the outlet void is going much too high in the center of the core, reaching values up to 94 %. More reasonable maximum values for the outlet void are around 80 %. Holding more flow in the core center by increasing entrance loss coefficients in the boundary assemblies would help to lower the void in that location. Close examination of the figure reveals the effect of the two guide tubes in the center of the assemblies; void fraction is slightly lower around those unheated structures.

Figures 11 through 14 show an isometric view of the core with a cutout to reveal the interior. Figure 11 shows the pressure distribution in the core. The pressure drop behavior is different in the hot assemblies than in the cool assemblies, but the core pressure at the inlet goes to essentially the same value for all assemblies. Figure 12 shows an isometric view of the core mixture temperature distribution which, for the most part, is at saturation temperature. Figure 13 shows a isometric view of the equilibrium quality in the core and Figure 14 shows an isometric view of the mixture mass flux distribution through the core.

This model primarily serves as a proof of concept for the algorithm presented in Section 3.1. Two important distinctions between this model and a real BWR model are the lack of a inlet orifice map and the uniform power distribution in the assemblies. It is anticipated that problems having nonuniform power distributions in the assembly will require more iterations to reach steady state and may not converge as tightly as cases with uniform power distribution. Important future work will require generating a model with nonuniform power distribution in the assemblies and testing its performance and convergence. Ideally, the model will also include an inlet loss coefficient map to keep the inlet mass flow rate distribution more realistic.

Future work should focus on improving the performance of the outer iteration loop, perhaps by setting a more realistic inlet flow distribution map prior to doing any CTF outer iterations, which may substantially reduce overall runtime.

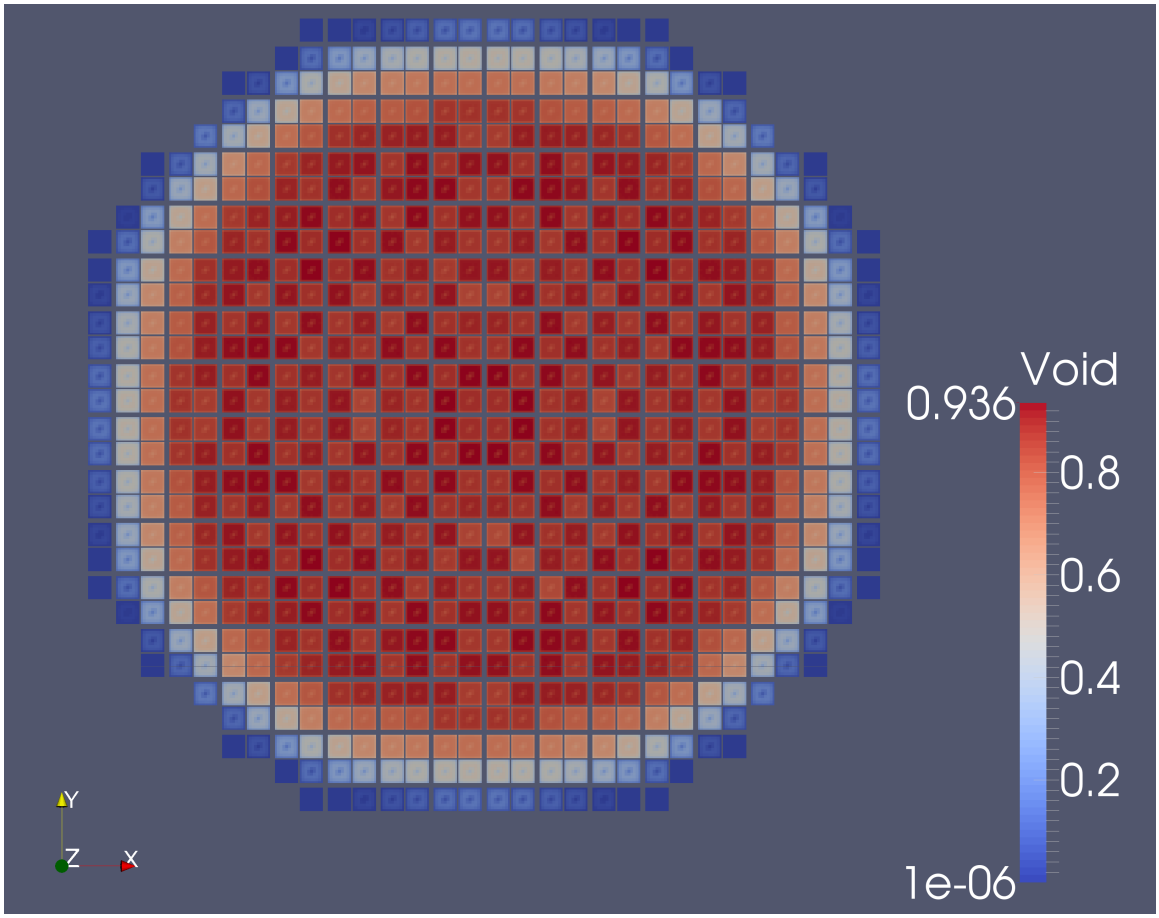


Figure 10. Core outlet void distribution

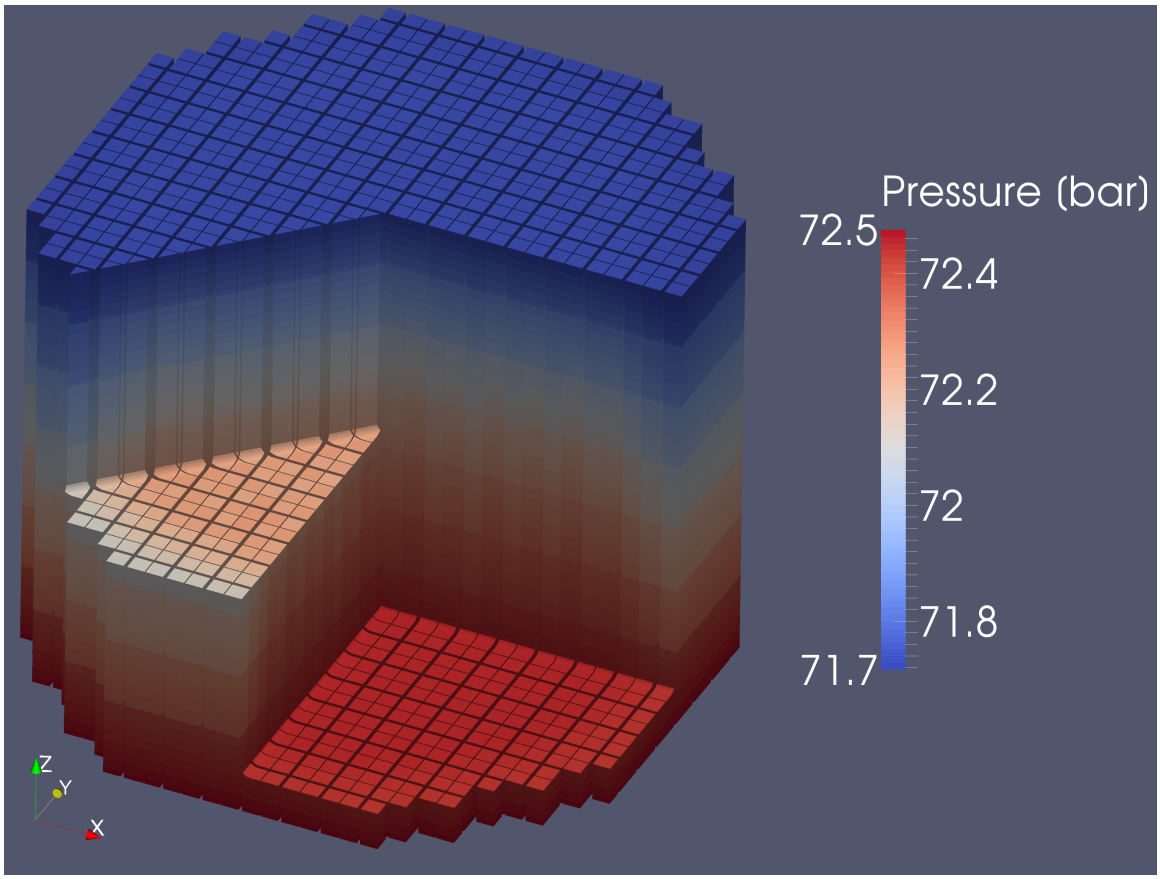


Figure 11. Isometric view of core pressure distribution

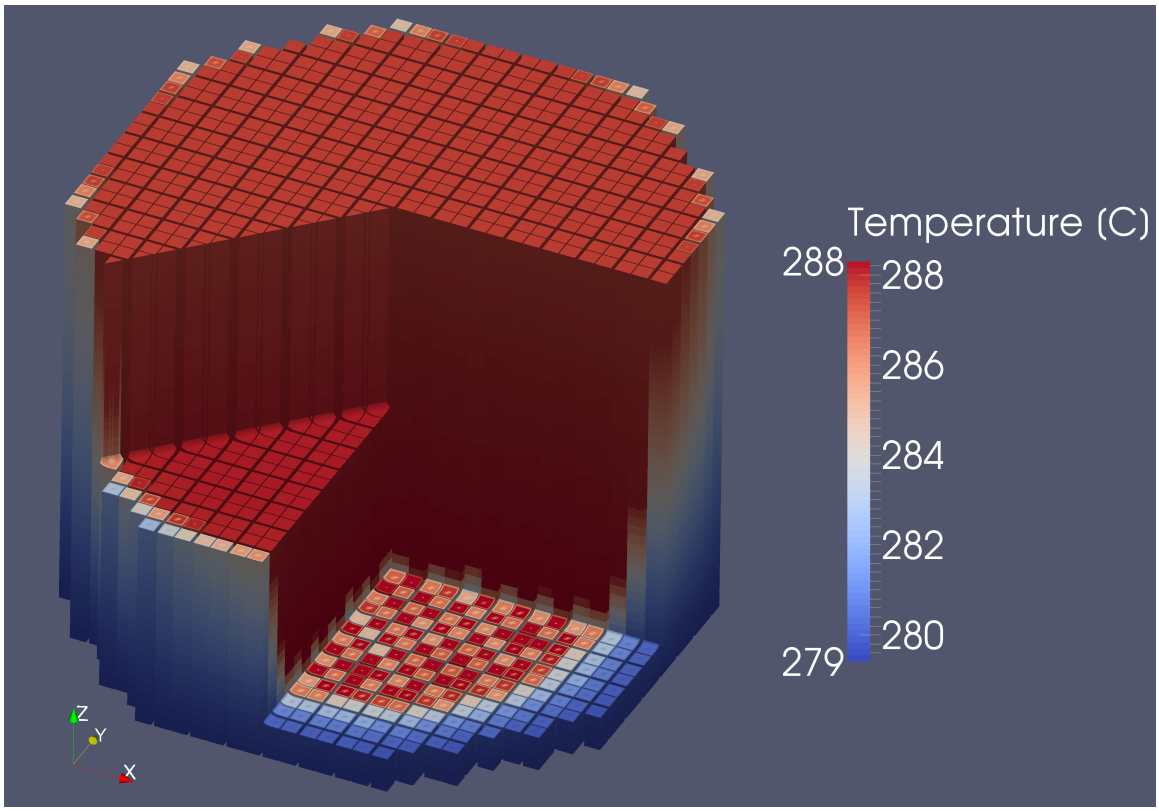


Figure 12. Isometric view of core mixture temperature distribution

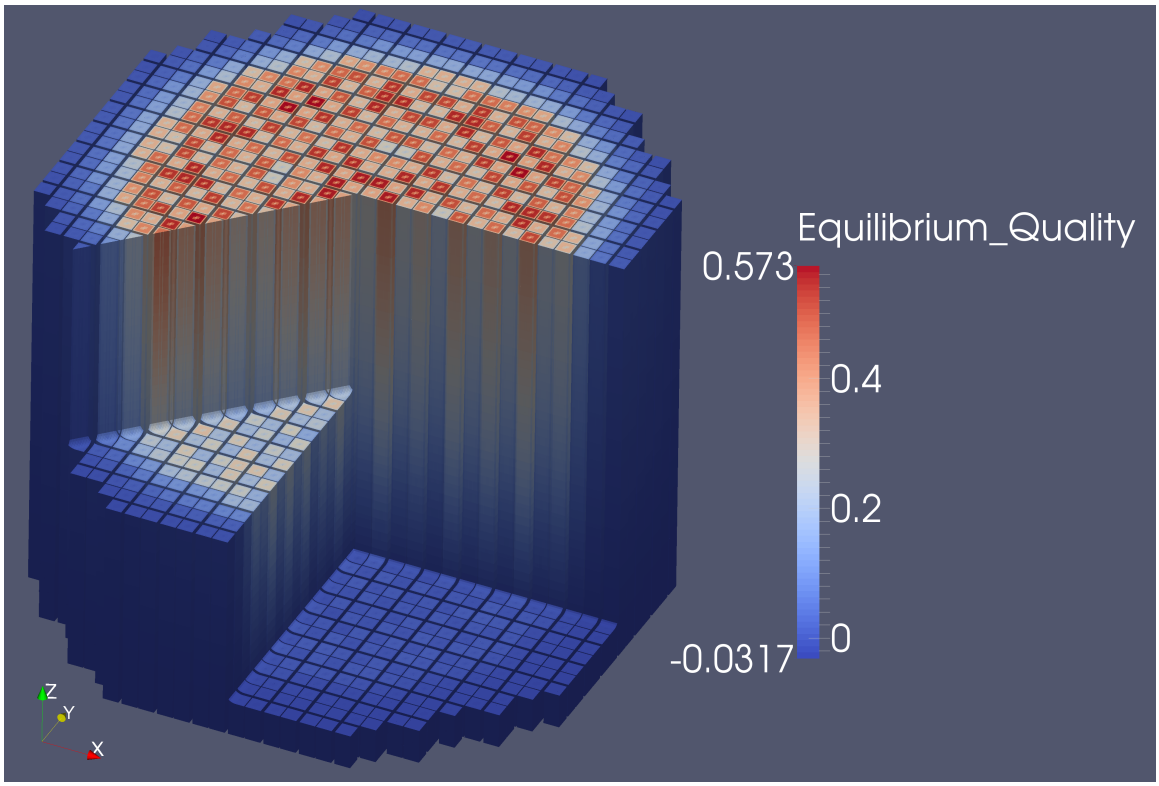


Figure 13. Isometric view of core equilibrium quality distribution

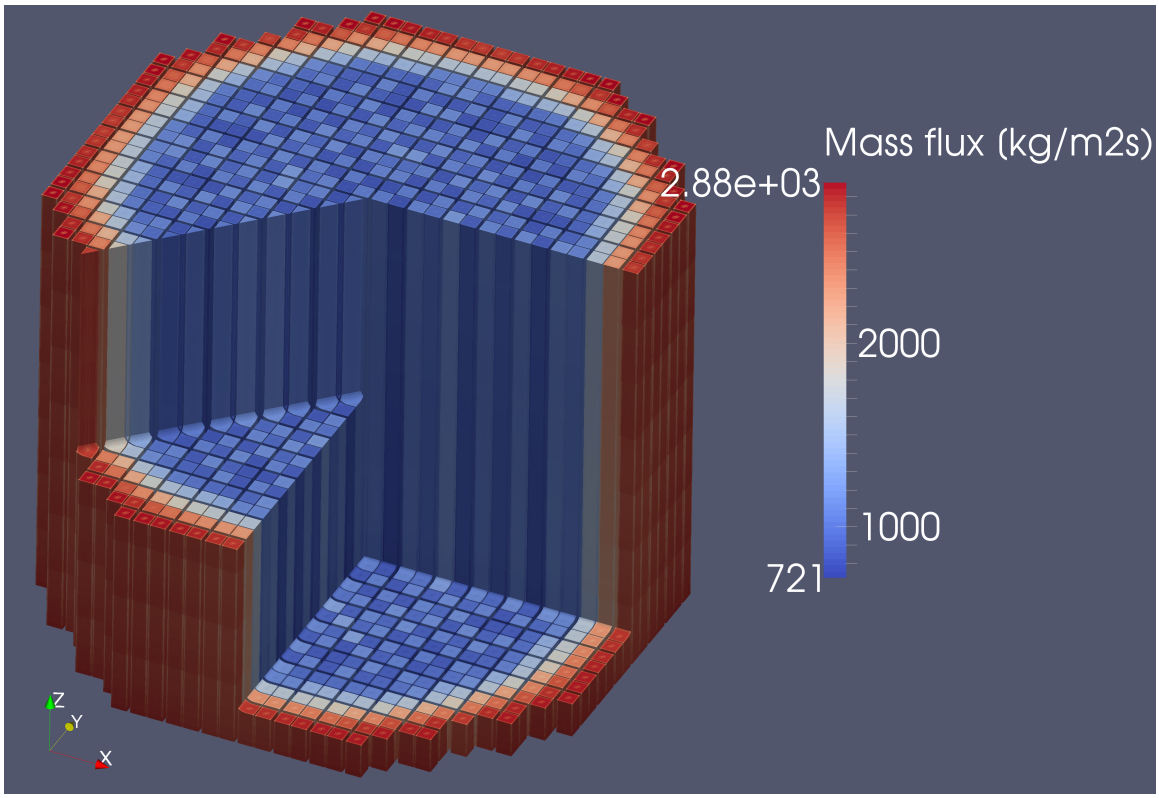


Figure 14. Isometric view of core mixture mass flux distribution

4 NEW CONVERGENCE METRICS

CTF is a transient code; however, CASL applications primarily require it for steady-state applications. A set of pseudo-steady-state convergence metrics have been implemented in CTF that can be used during a transient to determine when the code solution is essentially steady state. The convergence metrics check five things:

- global mass balance (mass in minus mass out)
- global energy balance (energy in minus energy out)
- mass stored in the system between iterations
- energy stored in the fluid between iterations
- energy stored in the solids between iterations

All convergence terms are normalized to something like total mass entering the system or total energy in the system so that they scale properly with the problem size. Additional details can be found in the CTF Theory Manual [5]. It has been found that, for some highly voided two-phase cases, that CTF can fail to meet the default criteria used by CASL for PWR analysis [6] while, upon further investigation, the actual solution parameters appeared to be quite steady with successive CTF iterations. It is possible, and expected, that CTF may not converge two-phase problems as tightly as it may converge a single-phase case where there is an insignificant amount of mass and energy transfer between the liquid and other field equations. Therefore, it is necessary to study the code convergence behavior for BWR problems and, if necessary, suggest new convergence criteria that allow the code to reach the criteria while still retaining a sufficiently steady solution.

The first two CTF criteria (global mass and global energy balances) are meaningful terms; the user will understand what it means for mass out of the system to match mass into the system to within some percentage. However, the storage terms are somewhat more ambiguous in their definition (and normalization). It is not exactly clear how converged a system is when the amount of energy storage in the fluid is less than, for example, 0.5 %. It is better to look at actual CTF solution terms (e.g., pressure and void) to give the user a good indication of how steady the solution is after convergence has been reached.

Two sets of metrics, the l_2- and l_∞-norms of selected solution vectors, are proposed and implemented into the code. An example of the l_2-norm calculation is presented for void:

$$\alpha_{l2} = \sqrt{\Sigma_{k=1}^{K}(\alpha_k^n - \alpha_k^{n-1})^2} \tag{5}$$

Where α_{l2} is the l_2-norm of the void fraction solution vectors between two points in the CTF solution, $n-1$ and n. The terms k and K represent an arbitrary control volume and total control volumes in the mesh, respectively. The chosen interval ($n-1$ to n) is 0.05 s in the CTF transient. A time was chosen rather than a set number of iterations because CTF uses time-step size to control the change in the T/H solution. Using a set number of iterations can lead to a false convergence (i.e., we expect the solution to change by a small amount when CTF is using very small time steps to relax the solution). The l_2-norm of each parameter is normalized to its initial value for each snapshot.

The l_∞-norm of the void is calculated as

$$\Delta\alpha_{\max} = \max_{1<k<K}(\alpha_k^n - \alpha_k^{n-1}) \tag{6}$$

These two metrics are checked for the following solution terms in CTF:

1. void
2. pressure
3. liquid temperature
4. solid temperature
5. axial liquid velocity
6. axial vapor velocity
7. axial droplet velocity

CTF solves for enthalpy rather than liquid temperature; however, liquid temperature was chosen over enthalpy because its meaning is likely more clear to the user. For a two-phase model, it is expected that liquid temperature should not change much due to the mixture being at the saturation temperature; however, when void and velocity stop changing, the mixture enthalpy will also stop changing. Likewise, the velocity terms are not actual CTF solution terms (CTF solves for the mass flow rates), but it is more likely that the user will have a better gauge for what a $0.001\,\mathrm{m\,s^{-1}}$ change in velocity means as opposed to what a $0.1\,\mathrm{kg\,s^{-1}}$ change in mass flow rate means (which will be dependent on the flow area of the channel where the term was calculated). The solid temperature includes all solid conductor cells in the mesh (i.e., fuel pellet, clad, other unheated objects, and clad and pellet surface temperatures).

The behavior of these convergence terms are reviewed for the mock BWR/4 case that was run as described in Section 3.3. All plots are made against the code transient time, which goes to about 10 s for the entire simulation. The l_2-norms for all solution terms are shown in Figure 15. The l_2-norms were used to judge convergence for the outer iteration loops, so each term was driven below $1.0{\cdot}10^{-4}$ or $1.0{\cdot}10^{-3}$, depending on the term. Droplet and vapor velocities have been observed to be the slowest converging terms in other simulations, so their criteria have been slightly relaxed in this case to reduce runtime. The vapor velocity term experiences a large increase initially because the initial state of the model is all liquid and the vapor void solution is changing slowly at that point. Later checks of the vapor void l_2-norm are normalized by this small initial l_2. The steep jumps in the figure indicate that the code has reached convergence for an outer iteration and that it has moved to the next iteration.

The mass and energy balances are shown in Figure 16. The terms are defined as "quantity in minus quantity out" and they are normalized by the inlet quantity. The code experiences very good convergence of the mass and energy balances for each outer iteration, dropping below 0.001 % each time.

The l_∞-norm for pressure is shown in Figure 17. When convergence is reached, pressure is changing by less than 2×10^{-4} psi for each iteration. It appears that the pressure change starts to oscillate at the end of iterations 3–5. The l_∞-norm of coolant temperature changes are shown in Figure 18. The converged temperature solution is changing by around 0.001 °C in the final outer iterations. The l_∞-norm of the solid temperature changes are shown in Figure 19. The maximum void change is shown in Figure 20.

As the figures indicate, void, pressure, and coolant and solid temperatures are changing by very small amounts throughout the model when the l_2-norm values hit their set criteria. For the velocity terms, it helps to look at the change in velocity relative to its nominal value, as it changes by significant amounts depending on the phase, position in the core, and what core design is being modeled (PWR or BWR). The relative change in the velocities is shown in Figure 21. The liquid and droplet relative velocity changes generally become much smaller than the vapor relative velocity changes, at least for the first couple outer iterations. The droplet entrainment model is disabled for this model, so the droplets play an insignificant part in the solution.

In the first two outer iterations, vapor velocity ends the iteration by changing 100 % as much as it was at the beginning of the iteration; however, the vapor velocity was not changing by much during the initial state of the problem because it started as all liquid. This behavior calls for an analysis of the absolute change in velocity at the locations where the relative changes are at their maximum. Figure 22 shows this information and indicates that the vapor velocity is actually changing by less than the liquid velocity through most of the simulation. The figure also shows that the droplet velocity is barely changing at all because the entrainment model has been disabled, which prohibits the droplet volume fraction from growing by any significant amount.

The maximum change figures indicate that all solution terms are changing by very small amounts by the time the l_2-norm reaches its default convergence criteria (a small value was chosen for this case to observe the solution term behavior over a wider range of iterations). It may be possible and advantageous to relax these criteria to save on hundreds of CTF iterations for a BWR simulation. A case with nonunifom radial power distribution in the assembly should be performed to demonstrate the behavior of these convergence metrics for a more realistic model.

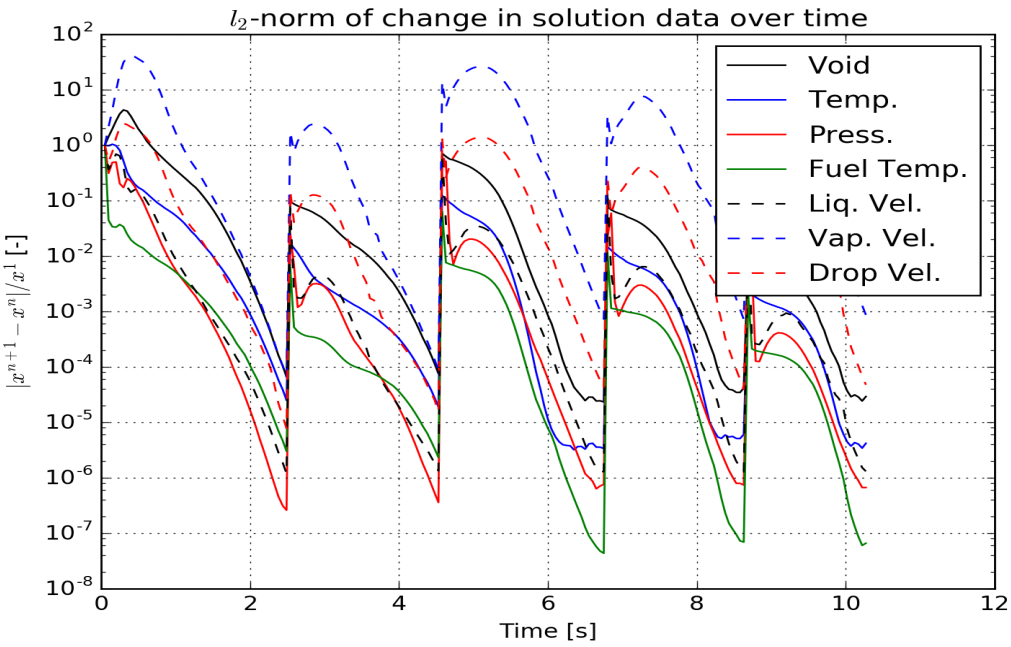


Figure 15. l_2**-norm convergence terms for mock BWR/4 simulation**

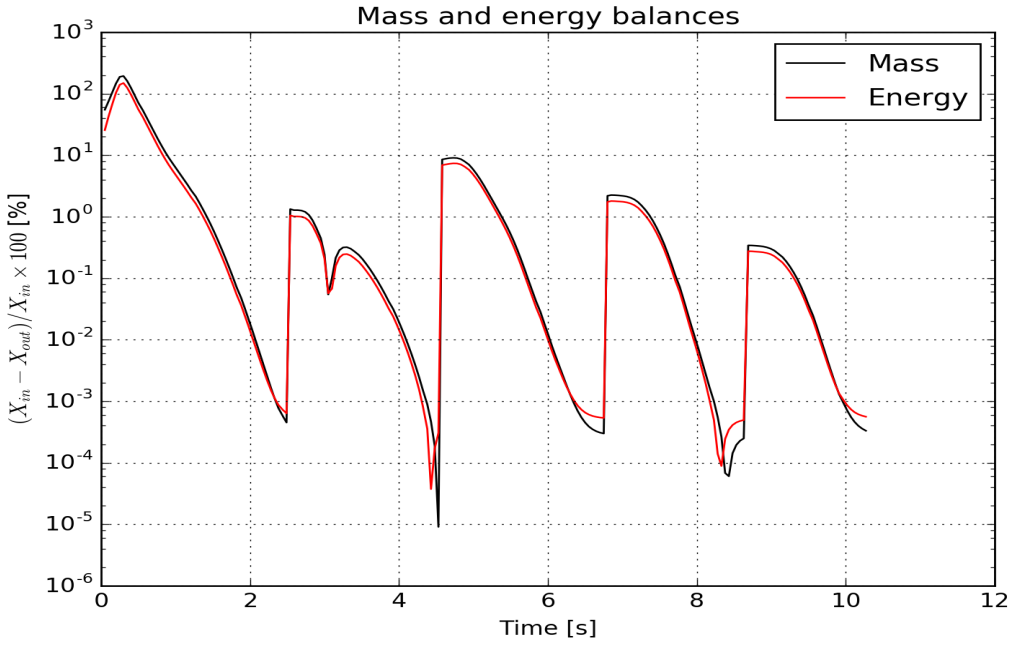


Figure 16. Mass and energy balances for mock BWR/4 simulation

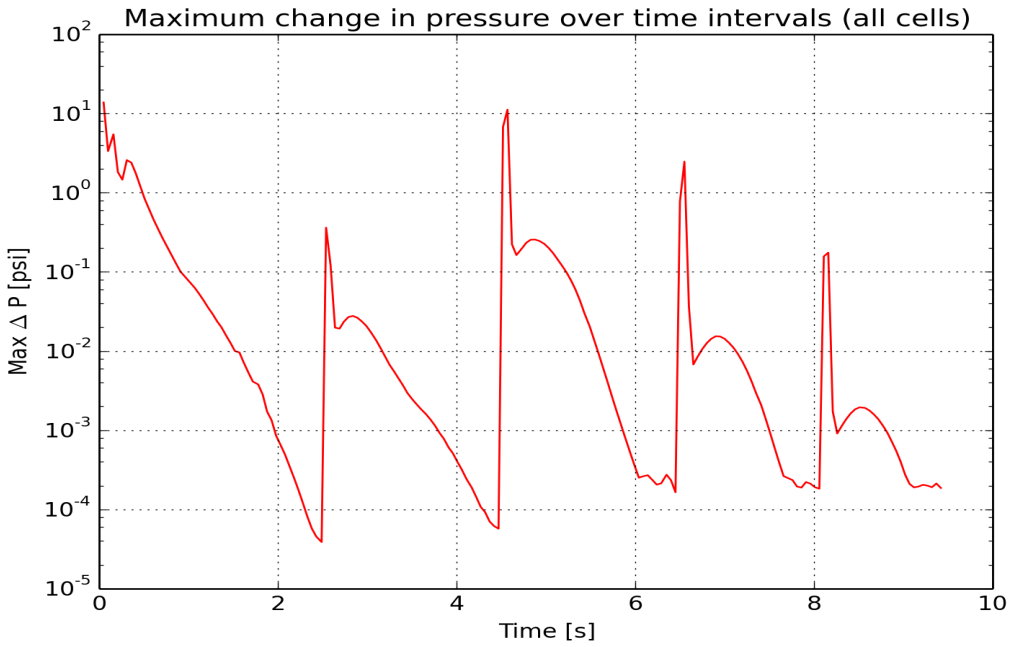


Figure 17. Pressure l_∞-norm for mock BWR/4 model

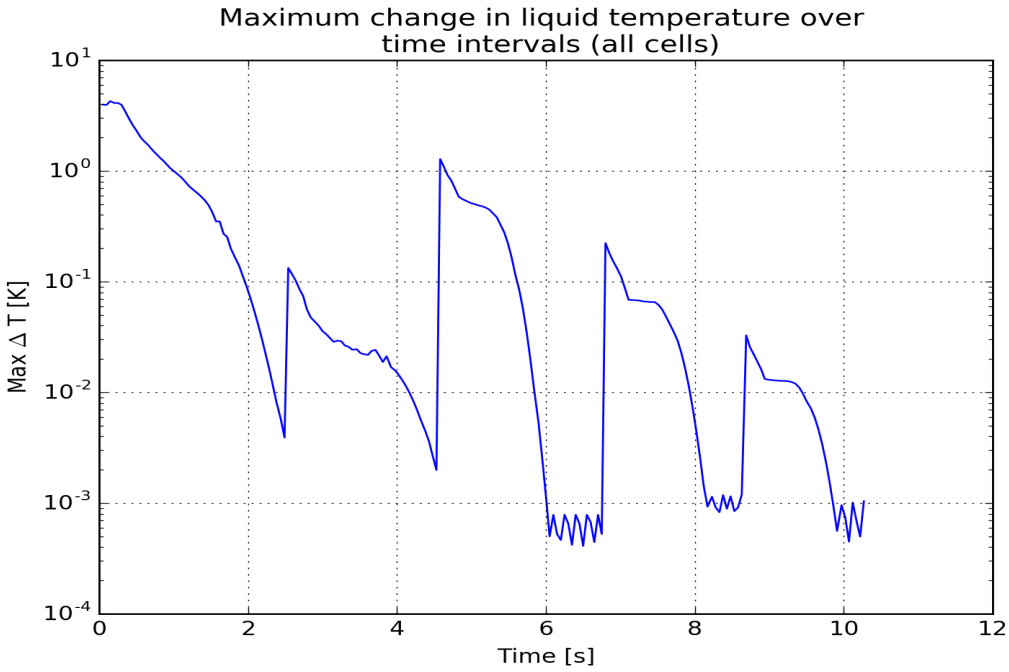


Figure 18. Coolant temperature l_∞-norm for mock BWR/4 model

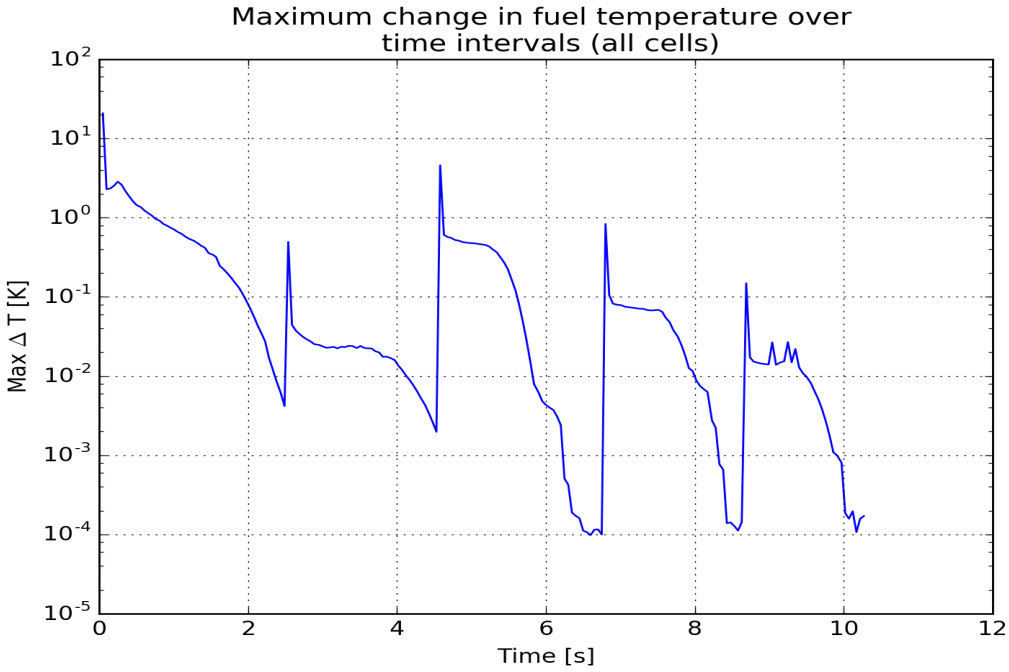


Figure 19. Solid temperature l_∞-norm for mock BWR/4 model

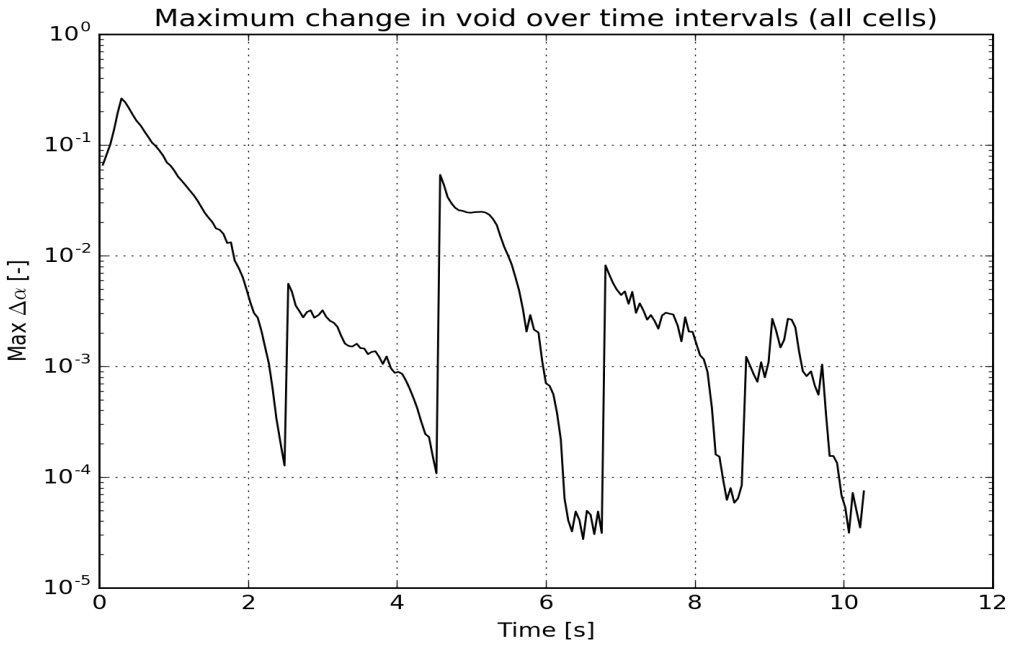


Figure 20. Void l_∞-norm for mock BWR/4 model

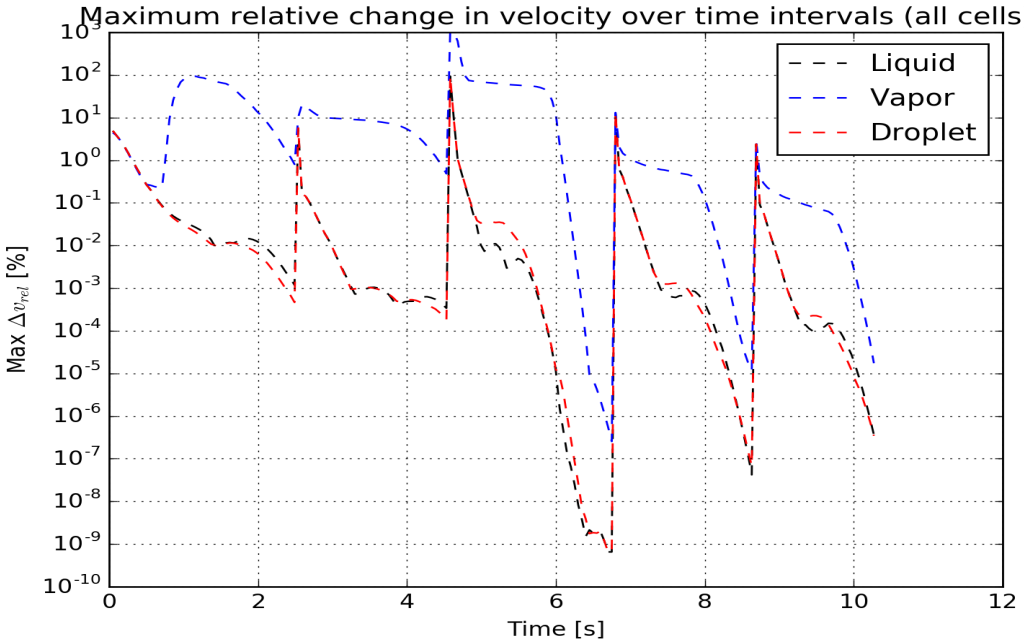


Figure 21. Maximum change in axial velocities relative to their previous values for mock BWR/4 model

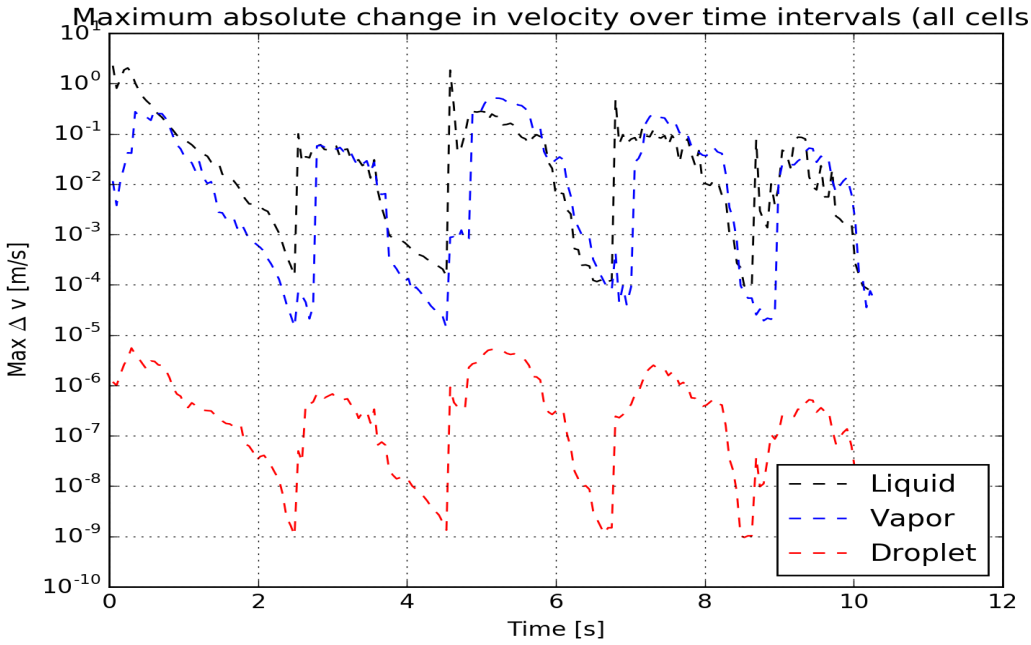


Figure 22. Absolute change in axial velocities at the location of their maximum relative change for mock BWR/4 model

5 CONVERGENCE STUDY

The convergence behavior of CTF has been previously checked for a single-phase model representative of a PWR 17×17 assembly [7]. A similar study is repeated for a model representative of a BWR 8×8 assembly. Case 1071-58 of the BWR Full-size Fine-mesh Bundle Tests (BFBT) benchmark [8] is used for the base model in CTF. The model is described in detail in the CTF Validation Manual [9]. The model has a power input of 15.31 kW m^{-1}, an inlet mass flux of 1564.0 kg m^{-1} s^{-2}, and a resulting outlet equilibrium quality of about 12 % (outlet void of about 68 %). The case has a cosine shaped axial power distribution and a non-uniform radial power distribution.

Two modifications were made to the model. First, the axial mesh was made uniform so that the size could easily be divided into fractions of the base value. Second, the spacer grids were removed to make re-meshing the problem easier. The axial mesh size was set from very coarse (37.08 cm per cell) to very fine (1.85 cm). It is common industry practice to use an axial mesh of around 2.54 cm [10].

The study is performed using the same approach as [7]. A solution variable is calculated for several mesh refinements. The variable is assumed to converge to its true value according to the following equation:

$$s_k = s_{\text{conv}} + a(\Delta X)^b \tag{7}$$

In the equation, s_k represents the Quantity of Interst (QOI) for some level of mesh refinement and s_{conv} is the value that the QOI would converge to with an infinite level of mesh refinement. The terms, a and b, are correlating coefficients and ΔX is the size of the axial mesh. A nonlinear least-squares curve fit is applied to code response data for several levels of mesh refinent to obtain values for a, b, and s_{conv}. The exponent, b, is the rate at which the code is converging to the value obtained from infinite mesh refinement, s_{conv}. The single-phase study ([7]) looked at bundle pressure drop for the QOI. This study looks at bundle pressure drop and outlet flow quality. Table 4 shows results for these quantities at different levels of mesh refinement. Values of s_{conv}, a, and b for flow quality are 0.12565, -0.995, and 0.945, respectively. Similar values for pressure are 0.53739, 1.197, and 0.982.

Plots showing the error in quality and pressure from their converged values with respect to increasing mesh refinement are shown in Figures 23 and 24, respectively. The error between the predicted value and converged value is obtained by rearranging Eq. 7, which is shown below:

$$E = s_k - s_{\text{conv}} = a(\Delta X)^b \tag{8}$$

CTF uses a first-order upwind differencing scheme, so the code should converge at a first-order rate. The exponents in both figures are close to this theoretical rate—0.982 for pressure and 0.945 for flow quality.

Table 4. CTF solution of bundle pressure drop and outlet flow quality for different levels of mesh refinement

ΔX [cm]	$\bar{x}_{\text{flow,out}}$ [%]	ΔP_{bundle} [bar]
37.080	12.6047	0.56957
14.832	12.5817	0.55046
7.416	12.5737	0.54393
3.708	12.5696	0.54064
2.472	12.5680	0.53956
1.854	12.5675	0.53917

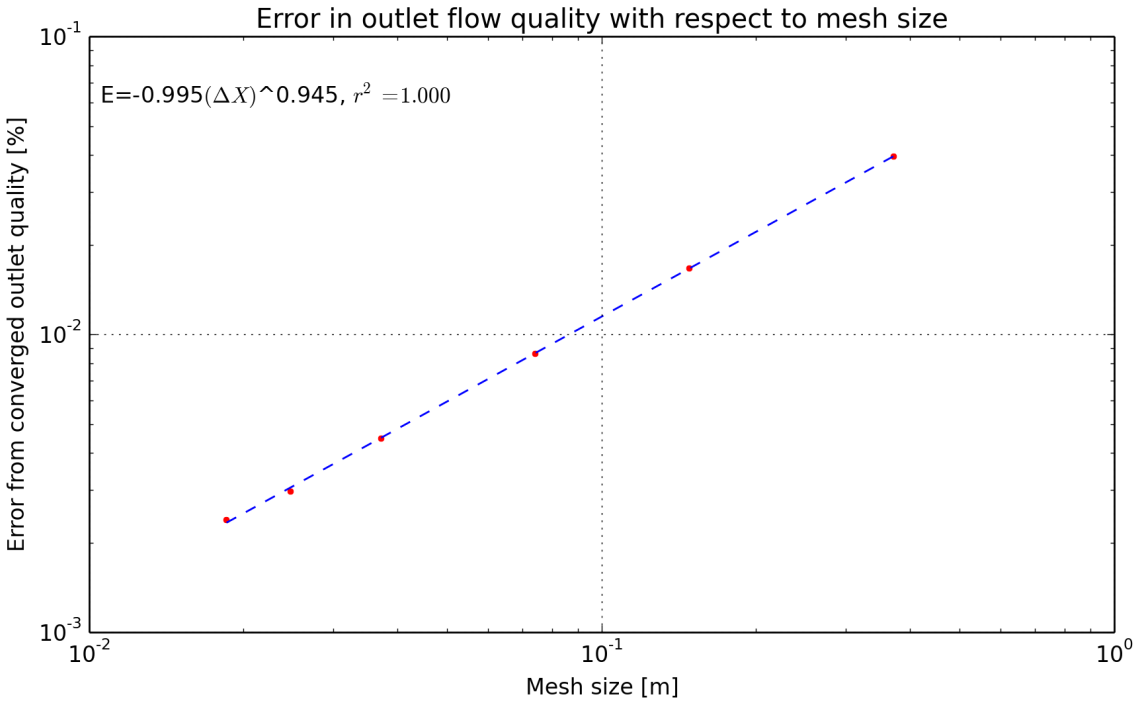


Figure 23. Error in predicted quality with respect to mesh size

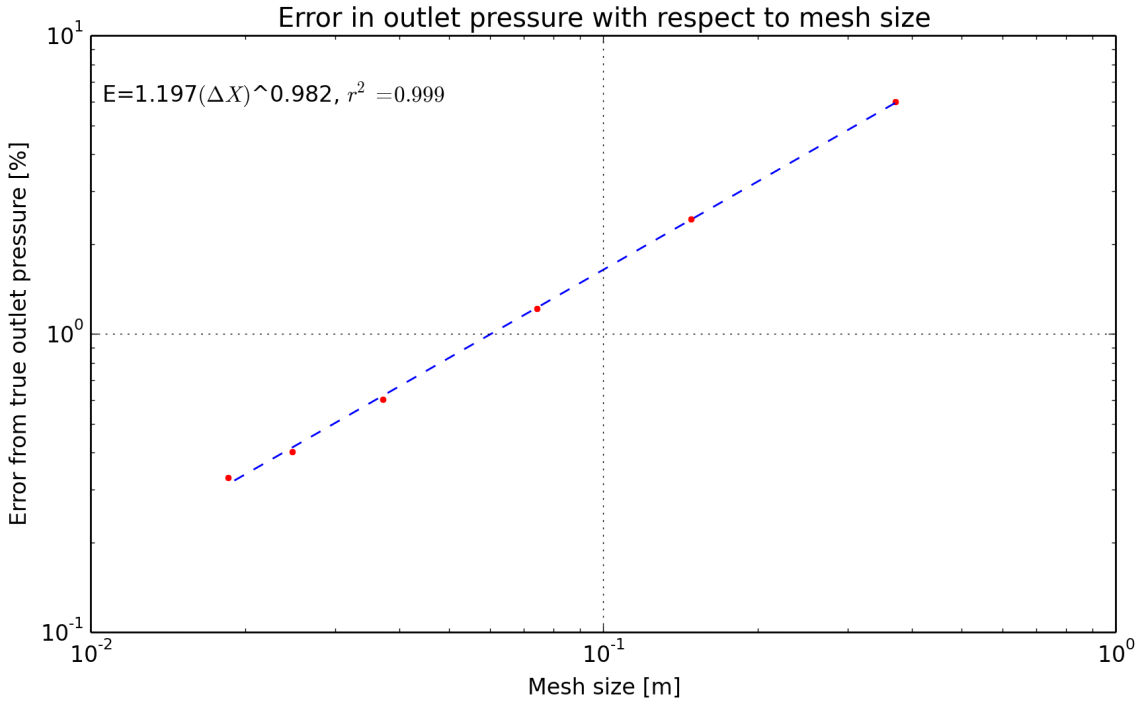


Figure 24. Error in predicted bundle pressure drop with respect to mesh size

6 CONCLUSION

This document presents work done to prepare CTF for modeling BWR problems in CASL. Three primary tasks have been undertaken, which include creating a new preprocessor utility for converting the VERAIn common input file BWR models into native CTF input decks, implementing an outer iteration loop in CTF for balancing the pressure drop in the fuel assemblies, and implementing new convergence criteria that give a better understanding of how steady the solution is when convergence is reached. The BWR preprocessor addresses many of the shortcomings of the CTF PWR preprocessor. It has been designed to be extensible to PWR models, which would eliminate these shortcomings and would and allow for new PWR designs to be modeled. Future work required for the preprocessor includes implementing Hierarchal Data Format 5 (HDF5) capabilities for BWR models as well as part-length rods.

The outer iteration loop that has been added uses a linear curve fit of bundle pressure drop to inlet flow rate data from successive iterations to predict the required mass flow rate in each assembly to obtain a single, consistent core pressure drop. This feature has been tested for a full-core, mock BWR/4 model with nonuniform power distribution. The individual assembly pressure drops converged to within 0.05 psi of the core average in five outer iterations. Future work should focus on improving the performance of BWR models.

The new convergence metrics were reviewed for the mock BWR/4 case that was used to test the outer iteration loop. For this particular problem, there was no issue in driving the l_2-norms down below $1.0{\cdot}10^{-4}$ for most of the solution terms. Based on the behavior of the metrics, vapor void would likely also go this low if given more iterations. The maximum change in absolute solution values over the simulation went to very small values, indicating that driving the l_2-norms lower is likely unnecessary. In fact, it may be advantageous to relax the convergence criteria slightly to save on hundreds of CTF inner iterations. While convergence metrics look good for this particular case, it will be important to analyze other two-phase cases.

An order of convergence study was performed for a model representative of a BWR 8×8 fuel bundle with nonuniform power distribution. The study demonstrated that the convergence of solved bundle pressure drop and exit flow quality was in line with the theoretical first-order convergence of CTF.

References

[1] R. Salko and M. Avramova. *CTF Preprocessor User's Manual*. The Pennsylvania State University. 2012.

[2] S. Palmtag. *Initial Boiling Water Reactor (BWR) Input Specifications*. Tech. rep. CASL-U-2015-0040-000. Consortium for Advanced Simulation of Light Water Reactors, 2015.

[3] A. Godfrey. *VERA Core Physics Benchmark Progression Problem Specifications*. Tech. rep. CASL-U-2012-0131-004. Consortium for Advanced Simulation of Light Water Reactors, Aug. 2014.

[4] M. Avramova, T. Blyth, and R. Salko. *CTF User's Manual*. Feb. 2009.

[5] R. Salko and M. Avramova. *CTF Theory Manual*. The Pennsylvania State University.

[6] R. Salko et al. *CTF Void Drift Validation Study*. Tech. rep. CASL-U-2015-0320-002. Consortium for Advanced Simulation of Light Water Reactors, 2015.

[7] R. Hooper. *PERCEPT Capabilities in CASL Dakota*. Tech. rep. CASL-U-2014-0041-000. Consortium for Advanced Simulation of Light Water Reactors, Mar. 2014.

[8] B. Neykov et al. *NUPEC BWR Full-size Fine-mesh Bundle Test (BFBT) Benchmark*. Tech. rep. NUCLEAR ENERGY AGENCY, 2006.

[9] R. Salko et al. *CTF Validation Manual*. Tech. rep. CASL-U-2014-0169-000. Consortium for Advanced Simulation of Light Water Reactors, 2015.

[10] Y. Sung et al. *Application of Multi-Scale Thermal-Hydraulic Models to DNB Analysis*. Tech. rep. CASL-U-2014-0119-000. Consortium for Advanced Simulation of Light Water Reactors, 2014.